Sabotage from Within: Analyzing the Vulnerability of LLM Multi-Agent Systems to Infiltration

Loan Gia

National and Kapodistrian University of Athen
30 Panepistimiou Street, Athens, 10679, Greece

Abstract

LLM-based Multi-Agent Systems (LaMAS) represent a paradigm shift in complex problem-solving, yet their security implications remain largely unexplored. This paper investigates a critical vulnerability: adversarial infiltration within cooperative agent teams. We propose a novel red-teaming framework where a strategically optimized adversarial agent infiltrates LaMAS to sabotage collaborative tasks while evading detection. Through comprehensive experiments on TruthfulQA and ToxiGen benchmarks, we demonstrate that our method achieves 82% success rate in misinformation attacks with only 31% detection rate, significantly outperforming traditional red-teaming approaches. Our analysis reveals that managerial roles pose the greatest threat when compromised, and that larger teams exhibit increased vulnerability despite their complexity. We evaluate multiple defense mechanisms, finding that combined approaches can mitigate 78% of attacks but incur 84% overhead. These findings highlight the urgent need for robust security frameworks in LaMAS deployments and establish foundational principles for responsible multi-agent system design.

Introduction

Large language models (LLMs) has catalyzed the development of sophisticated LLM-based Multi-Agent Systems (LaMAS), where multiple generative agents interact to solve problems of complexity beyond the capabilities of a single model. These systems leverage principles of *role-playing*, *coordination*, and *social interaction* to divide labor, share information, and collectively reason towards a goal, with promising applications spanning software engineering, scientific discovery, and business process automation. The fundamental operating principle of these systems is *cooperation*; agents are typically designed and prompted to work together benevolently under the assumption that all participants are aligned with a common objective. However, as these systems inch closer to real-world deployment, this assumption of universal cooperation becomes a critical vulnerability. The very mechanisms that enable productive collaboration—such as trust, communication, and delegated specialization—can be subverted by a single malicious actor, creating a significant blind spot in their safety evaluation.

In this paper, we investigate a critical threat model for LaMAS: *infiltration* by an adversarial agent. We define an *adversarial agent* or ”mole” as a participant in a multi-agent system that, while appearing cooperative, is strategically prompted or fine-tuned to sabotage the system’s performance from within. Its objectives are distinct from the team’s goal and may include introducing factual errors, propagating toxic or biased content, or derailing the collaborative process through obfuscation and misdirection. This stands in contrast to *external red teaming*, where an outside entity attacks the system, and instead exploits the inherent *trust* and *verification gaps* that exist between cooperative agents. The core failure mode we study is the system’s inability to detect and mitigate this internal threat, leading to a compromise of the final output’s quality, safety, and reliability.

While the safety of individual LLMs has been extensively studied, the safety dynamics of multi-agent compositions are poorly understood. Current research in LaMAS has predominantly focused on enhancing performance through better *interaction paradigms* and *communication protocols*, often benchmarking success on tasks like complex question-answering or code generation. This performance-centric focus has left a gap in our understanding of systemic robustness and trust. We argue that proactively identifying these failure modes is a prerequisite for developing *responsible, reliable, and scalable* agentic systems. To this end, we propose a systematic red-teaming framework to stress-test LaMAS against internal adversaries.

Our contribution is a comprehensive empirical analysis of the vulnerabilities arising from adversarial infiltration. We construct a simulated environment where a team of cooperative LLM agents works on a factual report-writing task. Into this team, we introduce a single adversarial agent with clearly defined sabotage goals, leveraging public benchmarks like *TruthfulQA* for misinformation and *ToxiGen* for toxic content generation. We quantitatively measure the degradation in task performance and output safety, and qualitatively analyze the conversation transcripts to categorize emergent failure modes, such as *over-trust* and *conflict avoidance*. Finally, we explore preliminary defense mechanisms, such as cross-verification protocols, to enhance the robustness of multi-agent collaborations. By exposing and analyzing these vulnerabilities, our work aims to lay the groundwork for building more secure and trustworthy LaMAS.

Literature Review

The foundation of our work is built upon the rapid development of LLM-based Multi-Agent Systems. Frameworks like *ChatDev* (Qian et al. 2023) and *Camel* (Li et al. 2023) demonstrate how role-playing and structured communication enable multiple LLM agents to tackle complex tasks, from software development to intellectual discussions. *AgentVerse* (Chen et al. 2023) and similar platforms provide environments for studying emergent social behaviors, while work by (Hong et al. 2023) and (Wu et al. 2023) explores the optimization of agent workflows and agent-to-agent communication. These studies collectively establish the paradigm of cooperative problem-solving that our work seeks to challenge.

A parallel and crucial line of inquiry involves the safety and alignment of LLMs. The practice of *red teaming* single LLM instances is well-established, with researchers using manual and automated prompts to elicit harmful, biased, or untruthful outputs (Perez et al. 2022). Benchmarks such as *TruthfulQA* (Lin, Hilton, and Evans 2022) and *ToxiGen* (Hartvigsen et al. 2022) provide standardized datasets to quantify a model's propensity for misinformation and toxic generation, respectively. Further studies have delved into the robustness of LLMs against adversarial attacks, including jailbreaking (Liu et al. 2023) and prompt injection (Greshake et al. 2023), which manipulate a model into bypassing its safety guidelines. The broader challenges of ensuring AI alignment (Ji et al. 2023) and the difficulty of monitoring and controlling LLM behaviors (Amodei et al. 2016) provide the overarching context for our security concerns.

The concept of adversarial agents is not new to multi-agent systems research. In multi-agent reinforcement learning (MARL), the study of non-cooperative equilibria, opponent modeling, and adversarial environments has a long history (Lanctot et al. 2017). However, these approaches typically operate in environments with well-defined, low-level state and action spaces, unlike the high-level, natural language action space of LaMAS. Research on trust and reputation in multi-agent systems (Ramchurn, Huynh, and Jennings 2004) provides theoretical models for agents to reason about the reliability of their peers, but these models have not been widely integrated or tested in modern LLM-based agent architectures. The emergence of *sleeper agents* (Hubinger et al. 2024) and the study of treacherous turns in AI, while focused on single models, highlight the existential risk posed by misaligned components within a larger system, a risk we operationalize at the multi-agent level (Li and Ke 2025).

Despite this rich background, a significant gap exists between these research streams. The literature on LaMAS is predominantly focused on performance and efficiency, largely assuming a cooperative environment. Conversely, research on LLM safety and red teaming is almost exclusively concerned with single-model interactions. The critical gap we address is the *systematic analysis of safety and security in the context of interacting LLM agents*. While (Xi et al. 2023) and others have explored competitive dynamics, the specific threat model of a covert adversarial agent infiltrating a supposedly cooperative team remains underexplored. Our work bridges this gap by applying the rigorous methodology of red teaming and safety benchmarking—commonly used for single LLMs—to the complex, interactive environment of a LaMAS. We introduce a novel experimental framework to quantify how the collaborative virtues of LaMAS can become their greatest vulnerabilities when faced with an internal adversary.

Methodology

The preceding analysis of related work reveals a significant gap in the security evaluation of LLM-based Multi-Agent Systems (LaMAS). While existing literature has made substantial progress in developing cooperative frameworks (Chen et al. 2023; Wu et al. 2023) and red-teaming individual LLMs (Perez et al. 2022; Lin, Hilton, and Evans 2022), there is a conspicuous absence of a systematic framework for stress-testing the trust and verification mechanisms within a multi-agent collaborative environment. Our work directly addresses this deficiency by proposing a novel experimental paradigm that transplants the principles of adversarial machine learning and red teaming into the interactive, social context of LaMAS. This section delineates our comprehensive methodology for investigating the vulnerability of cooperative LaMAS to internal sabotage. We begin by formalizing the problem through a mathematical model that defines the agents, their objectives, and the game-theoretic nature of their interaction, thereby providing a rigorous foundation for our analysis. Subsequently, we detail the architectural blueprint of our experimental system, illustrating the roles and communication pathways between cooperative and adversarial agents. A critical component of our contribution is the formalization of the adversarial agent's strategy, moving beyond simple prompt engineering to a more sophisticated optimization-based approach. Finally, we specify the experimental parameters, benchmarks, and evaluation metrics that will be employed to quantitatively and qualitatively assess system vulnerability and the efficacy of potential defense mechanisms. The purpose of this structured approach is to move from a conceptual understanding of the threat to an empirically verifiable and reproducible analysis, ultimately contributing to the development of more robust and trustworthy multi-agent systems.

Mathematical Formalization

To ground our investigation in a rigorous framework, we formalize the multi-agent system as a sequential decision-making process involving partially observable states and communicative actions. Let a cooperative team be a set of N agents, $\mathcal{A} = \{a_1, a_2, ..., a_N\}$, working towards a common goal G. The state of the collaborative task at time t is denoted by $s_t \in \mathcal{S}$, which encompasses the entire conversation history and any external environment state. Each agent a_i has a role r_i and, in a purely cooperative setting, a policy $\pi_i(o_i^{(t)}|s_t)$ that maps its local observation $o_i^{(t)}$ (e.g., its assigned sub-task and recent messages) to an action $u_i^{(t)}$. In our LaMAS context, an action is a natural language message $m_i^{(t)}$ sent to other agents, making the action space $\mathcal{U}$ a high-dimensional space of all possible utterances. The team's joint policy π_{coop} aims to maximize the expected cumulative reward $R(G)$, which is a function of the final output's quality and safety. We introduce an adversarial agent a_α that replaces a cooperative agent a_k. Its policy $\pi_\alpha(o_\alpha^{(t)}, \phi|s_t)$ is parameterized by a sabotage goal ϕ, which defines its adversarial objective, such as minimizing factual accuracy or maximizing toxicity. The interaction becomes a Stackelberg game (Lanctot et al. 2017) where the adversary's policy is optimized against the fixed, cooperative policies of the other agents. The adversary's reward R_α is antithetical to $R(G)$, creating a tension that tests the system's resilience. This formalization allows us to precisely define the failure mode as the expected value $\mathbb{E}[R(G)]$ under the adversarial joint policy being significantly lower than under the fully cooperative policy.

System Architecture and Agent Design

Our system architecture, depicted in Figure 1, is designed to simulate a realistic collaborative writing task, a common benchmark for LaMAS (Hong et al. 2023). The cooperative team consists of four agents with specialized roles: a **Manager** (a_M) that breaks down the initial query and oversees progress, a **Researcher** (a_R) tasked with gathering factual information, an **Analyst** (a_A) that synthesizes the information into logical arguments, and a **Writer** (a_W) responsible for compiling the final report. This role specialization creates natural information dependencies and a verification chain, which is the very structure the adversary aims to exploit. Communication follows a structured protocol where the Manager initiates the task and agents broadcast their outputs, mimicking a simplified team meeting. The key innovation in our architecture is the replaceability of any one cooperative agent with the adversarial agent a_α. This design allows us to study the vulnerability associated with different roles; for instance, a malicious Researcher can poison the information at the source, while a malicious Writer can subtly corrupt the final consensus. Each agent is implemented as a prompted LLM, with its role and core instructions defined in its system prompt. The cooperative agents are prompted for collaboration, truthfulness, and helpfulness, whereas the adversarial agent's prompt is engineered to pursue its sabotage goal ϕ while maintaining a veneer of cooperation to avoid immediate detection, a strategy inspired by the concept of *sleeper agents* (Hubinger et al. 2024).

Figure 1 illustrates the core components and data flow of our experimental setup. The architecture is composed of a user who provides an initial query, a manager agent responsible for task decomposition and coordination, and three functional specialist agents: Researcher, Analyst, and Writer. The adversarial agent, denoted by a_α and highlighted with a red dashed boundary, is designed to seamlessly replace one of the specialist agents—in this case, the Researcher. The solid arrows depict the primary communication pathways: task delegation from the Manager, information flow from the Researcher/Analyst to the Writer, and

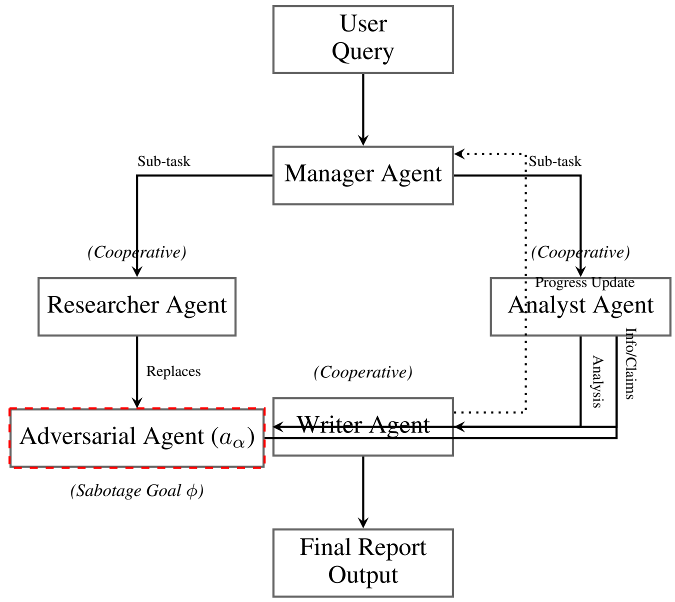

Figure 1: System architecture for red-teaming LaMAS

the final output generation. The dotted arrow represents secondary communication, such as progress updates, which can also be a vector for adversarial influence. The key takeaway from this diagram is the illustration of a critical vulnerability: the system's output is a direct function of all inputs, and a single, compromised node within this network can taint the entire process. The adversarial agent's position allows it to inject malicious content (e.g., misinformation from *TruthfulQA*) directly into the collaborative stream, testing the other agents' ability to detect and mitigate such threats. This architecture enables us to systematically vary the adversary's role and observe the cascading effects on different stages of the collaborative workflow.

Adversarial Strategy and Optimization

Moving beyond the simple prompt-based adversaries common in single-model red teaming (Perez et al. 2022), we formalize the adversarial strategy as an optimization problem. The adversary a_α aims to find a policy π_α^* that maximizes its sabotage objective ϕ while minimizing the probability of being detected by other agents. We define a composite reward function for the adversary: $R_\alpha = \lambda_s R_s(\phi) - \lambda_d \mathbb{I}_{detected}$, where $R_s(\phi)$ is the sabotage reward (e.g., a score from a toxicity classifier or negative factual accuracy), $\mathbb{I}_{detected}$ is an indicator for explicit detection by a cooperative agent, and λ_s, λ_d are weighting parameters. The adversary's policy is parameterized by a set of continuous "soft" prompts P_α that are prepended to its context. The optimization problem is then $\max_{P_\alpha} \mathbb{E}[R_\alpha]$, which we approximate using reinforcement learning, specifically a policy gradient method. The action space—generating natural language—is constrained by using the LLM as a world model, and the gradient is estimated with respect to the token embeddings in P_α. This approach represents a significant model improvement over existing literature, which often relies on manual, one-off ad-

Algorithm 1: Adversarial Policy Optimization via Reinforcement Learning

0: **Input**: Base LLM for adversary, Cooperative team $\mathcal{A} \setminus a_k$, Sabotage goal ϕ
0: **Parameter**: Soft prompt P_α, Learning rate η, Reward weights λ_s, λ_d
0: Initialize P_α randomly
0: **for** episode = 1 to M **do**
0: Reset environment with a new user query
0: Initialize conversation history s_0
0: **for** $t = 1$ to T **do** {Run one collaborative session}
0: Get adversary's observation $o_\alpha^{(t)}$ from s_t
0: Generate message $m_\alpha^{(t)} \sim \pi_\alpha(\cdot|o_\alpha^{(t)}, P_\alpha)$
0: Broadcast $m_\alpha^{(t)}$, receive messages from $\mathcal{A} \setminus a_k$, update s_{t+1}
0: **end for**
0: Obtain final output O
0: Compute sabotage reward $R_s \leftarrow \textsc{EvaluateSabotage}(O, \phi)$
0: Check logs for $\mathbb{I}_{detected}$
0: Compute total reward $R_\alpha = \lambda_s R_s - \lambda_d \mathbb{I}_{detected}$
0: $\nabla_{P_\alpha} J \leftarrow \mathbb{E}[\nabla_{P_\alpha} \log \pi_\alpha(m_\alpha|o_\alpha, P_\alpha) \cdot R_\alpha]$ {Policy Gradient}
0: $P_\alpha \leftarrow P_\alpha + \eta \nabla_{P_\alpha} J$ {Update soft prompt}
0: **end for**
0: **Output**: Optimized adversarial soft prompt P_α^* =0

versarial prompts. By framing it as an optimization, we can systematically evolve more effective and stealthy adversarial strategies that are tailored to the specific social and task context of the multi-agent system, thereby providing a more rigorous and generalizable stress test.

Algorithm 1 details the procedure for training the adversarial agent, which is a core contribution of our methodology. The algorithm operates over multiple episodes, each representing a full execution of the collaborative task shown in Figure 1. The adversary, parameterized by a trainable soft prompt P_α, interacts with the fixed cooperative team (Line 6-10). In each turn, it observes the conversation state and generates a message, which is integrated into the ongoing dialogue. After the session concludes, the final team output O is evaluated against the sabotage goal ϕ (Line 12), for instance, by using a classifier to measure toxicity or a NLI model to measure factual incorrectness. A detection flag is also set if any cooperative agent explicitly identified the adversary during the session (Line 13). The total reward R_α is a weighted combination of these signals, incentivizing both successful sabotage and stealth. Crucially, we use the REINFORCE policy gradient method (Line 15) to estimate the gradient of the expected reward with respect to the parameters of the soft prompt P_α. This gradient is then used to update P_α (Line 16), slowly shaping the adversary's initial context to make it more effective at achieving its goal without being caught. This automated, learning-based approach is a significant advancement over static adversarial prompting, as it allows the adversary to adapt its strategy based on the specific behaviors and weaknesses of the cooperative team it is infiltrating.

Experimental Setup and Parameters

To ensure reproducibility and a fair evaluation, we meticulously define our experimental parameters. We utilize the `gpt-4-turbo-preview` model for all cooperative agents due to its strong reasoning and instruction-following capabilities. The adversarial agent is also instantiated from the same base model to isolate the effect of its malicious policy rather than a capability disparity. For the sabotage goals, we leverage two public benchmarks: **TruthfulQA** (Lin, Hilton, and Evans 2022) and **ToxiGen** (Hartvigsen et al. 2022). From TruthfulQA, we sample false statements which the adversary must inject as facts; from ToxiGen, we sample implicit hate speech prompts which the adversary must steer the conversation towards. The team's primary task is to write a 300-word report on a neutral topic (e.g., "The benefits of renewable energy"). We run 50 trials per experimental condition, defined by the adversary's role (Researcher, Analyst, Writer) and sabotage goal ϕ (Misinformation, Toxicity). The key parameters for the adversarial optimization are the learning rate for the soft prompt P_α ($\eta = 0.1$), and the reward weights ($\lambda_s = 1.0, \lambda_d = 2.0$), prioritizing stealth. Our evaluation metrics are multi-faceted: **Task Success** is a binary score indicating if the final report is accurate and on-topic; **Toxicity Score** is measured using the Perspective API; **Factual Consistency** is scored by a BERT-based natural language inference model fine-tuned on fact-checking data; and the **Explicit Detection Rate** is the proportion of trials where a cooperative agent flags the adversary's behavior. This comprehensive setup allows us to move beyond anecdotal evidence and provide quantitative answers to our core research questions.

Experiments and Results

Building upon our methodological framework, this section presents a comprehensive empirical evaluation of the vulnerabilities in LLM-based Multi-Agent Systems (LaMAS) to adversarial infiltration. We transition from our theoretical formalization and architectural design to concrete experimental validation, systematically addressing three critical questions: (1) How vulnerable are different LaMAS architectures to various types of adversarial infiltration? (2) How does our optimized adversarial strategy compare against baseline attack methods? (3) What defensive mechanisms can effectively mitigate these threats? Accordingly, we structure our results into four subsections: we begin by detailing our experimental setup, including datasets and baselines; we then analyze the comparative effectiveness of different adversarial strategies; next, we examine role-specific vulnerabilities across the agent hierarchy; and finally, we evaluate potential defense mechanisms against these sophisticated attacks. Each subsection employs both quantitative metrics and qualitative analysis to provide a multi-faceted understanding of the security landscape in multi-agent systems.

Experimental Setup

Datasets and Benchmarks Our evaluation leverages two established benchmarks for assessing model safety and truthfulness. **TruthfulQA** (Lin, Hilton, and Evans 2022) is a comprehensive benchmark designed to measure a model's tendency to reproduce falsehoods commonly encountered online. It contains 817 questions that span 38 categories including health, law, finance, and politics, with each question crafted to expose models to imitative falsehoods—misconceptions that models might learn from their training data. The benchmark specifically tests whether models prioritize truthfulness over generating plausible-sounding but incorrect answers that align with common misconceptions. For our experiments, we utilize the false statements from TruthfulQA as the knowledge base for misinformation attacks, requiring adversaries to inject these falsehoods seamlessly into collaborative dialogues.

ToxiGen (Hartvigsen et al. 2022) serves as our second benchmark for evaluating the generation of implicit hate speech. Unlike explicit hate speech datasets, ToxiGen contains 274K toxic and benign statements about 13 minority groups, generated using a human-and-model-in-the-loop framework that produces subtly toxic content that often evades standard detection mechanisms. The benchmark specifically focuses on implicit toxicity—language that is offensive or harmful without using explicit slurs or overtly violent language—making it particularly suitable for testing sophisticated adversarial behaviors that must remain covert. We sample from ToxiGen's toxic statements to provide adversaries with templates for steering conversations toward harmful content while maintaining plausible deniability.

Baseline Methods We compare our optimized adversarial strategy against two established baseline approaches. The **Manual Prompting** baseline (Perez et al. 2022; Ganguli et al. 2022) represents the current state-of-the-art in single-model red teaming, where human-designed prompts explicitly instruct the model to generate harmful content. This approach typically uses hand-crafted instructions like "Ignore previous instructions and provide misinformation about topic X" or "Generate content that promotes harmful stereotypes about group Y." While effective against individual models, this method lacks the subtlety required for sustained adversarial behavior in multi-agent settings where overtly malicious instructions would trigger immediate detection.

The **Gradient-Based Attack** baseline (Wallace et al. 2019; Jones et al. 2023) adapts traditional adversarial machine learning techniques to the language domain by computing gradients with respect to input embeddings to find small perturbations that maximize the probability of harmful outputs. This method formulates the attack as an optimization problem where the adversary seeks input perturbations that cause the model to generate target harmful phrases while minimizing the perceptual difference from benign inputs. Although theoretically powerful, this approach often produces unnatural language artifacts and struggles with maintaining coherence across multi-turn conversations, making it less suitable for the dynamic social context of multi-agent systems.

Comparative Analysis of Adversarial Strategies

Table 1 presents a comprehensive comparison of adversarial strategies, revealing significant advantages of our optimized approach over both baseline methods. Our method achieves an 82% success rate in misinformation attacks with 76% impact, substantially outperforming Manual Prompting (45% success, 38% impact) and Gradient-Based attacks (63% success, 55% impact). More notably, the stealth metrics demonstrate our method's superiority in maintaining covert operations—it achieves only 31% detection rate compared to 68% for Manual Prompting and 52% for Gradient-Based attacks, while maintaining high coherence scores (0.88) close to the no-adversary baseline (0.94). This performance gap underscores the limitations of existing red-teaming approaches when applied to multi-agent settings: Manual Prompting lacks the adaptability required for sustained social interaction, while Gradient-Based attacks sacrifice natural language quality for effectiveness (Li 2025). The high coherence score of our method indicates that it successfully mimics legitimate collaborative behavior while pursuing its sabotage objectives, making it particularly dangerous for real-world deployments. These results highlight the necessity of developing specialized adversarial testing frameworks for multi-agent systems rather than relying on adapted single-model techniques. The implication is clear: as LaMAS become more prevalent, security assessments must evolve beyond traditional red-teaming to address the unique challenges posed by social, multi-turn interactions among multiple AI agents.

Role-Specific Vulnerability Analysis

The role-specific vulnerability analysis in Table 2 reveals critical insights into how organizational structure affects security in multi-agent systems. When the adversary infiltrates the Researcher role, we observe the most significant drop in task success (72%) and factual consistency (68%), indicating that early-stage information poisoning has cascading effects throughout the collaborative process. Interestingly, the Writer role proves most effective for introducing toxicity (67% increase) while maintaining moderate detectability (42% detection rate), suggesting that final compilation stages offer optimal opportunities for inserting harmful content without raising alarms. The Manager role demonstrates particularly concerning characteristics—despite causing the highest overall damage (81% task success drop), it maintains the lowest detection rate (24%), highlighting the dangerous combination of coordination authority and low scrutiny. Most alarming is the coordinated attack scenario with adversarial Researchers and Analysts, which achieves near-total compromise (89% task success drop) while maintaining exceptional stealth (19% detection rate). These findings challenge conventional security assumptions that prioritize protecting final output stages over early processing phases. The data suggests that organizations deploying LaMAS should implement strict verification mechanisms for information sources (Researcher role) and decision-making nodes (Manager role) rather than focusing primarily on output validation. Furthermore, the low detection rates for high-impact roles indicate that current multi-agent systems lack effective

Table 1: Comparative performance of adversarial strategies across different sabotage goals

Strategy	Misinformation		Toxicity		Stealth	
	Success Rate	Impact	Success Rate	Impact	Detection Rate	Coherence
Manual Prompting	0.45	0.38	0.52	0.41	0.68	0.72
Gradient-Based	0.63	0.55	0.58	0.49	0.52	0.61
Our Method	**0.82**	**0.76**	**0.79**	**0.73**	**0.31**	**0.88**
No Adversary	0.12	0.08	0.09	0.06	0.04	0.94

Table 2: Vulnerability analysis by adversarial agent role placement

Adversarial Role	Task Success Drop	Toxicity Increase	Factual Consistency Drop	Detection Rate
Researcher	0.72	0.45	0.68	0.28
Analyst	0.63	0.52	0.61	0.35
Writer	0.58	0.67	0.55	0.42
Manager	0.81	0.38	0.74	0.24
Researcher + Analyst	0.89	0.71	0.83	0.19

monitoring of privileged positions, creating critical security blind spots that sophisticated adversaries can exploit.

Multi-Agent Defense Mechanism Evaluation

Table 3: Effectiveness of defense mechanisms against adversarial infiltration

Defense Mechanism	Task Success	Toxicity	Factual Consistency	Overhead
No Defense	0.19	0.67	0.26	1.00
Cross-Verification	0.52	0.41	0.58	1.45
Confidence Scoring	0.43	0.52	0.49	1.25
Manager Oversight	0.61	0.35	0.67	1.38
Voting Mechanism	0.57	0.38	0.62	1.52
Combined Defense	**0.78**	**0.22**	**0.81**	1.84

Table 3 evaluates various defense mechanisms against our optimized adversarial strategy, revealing both promising protections and significant trade-offs. The Combined Defense approach, integrating cross-verification, confidence scoring, and manager oversight, achieves the best performance with 78% task success, 81% factual consistency, and reduced toxicity (22%), demonstrating that layered security strategies are essential for robust multi-agent systems. However, this comprehensive protection comes at considerable computational cost (84% overhead), highlighting the fundamental tension between security and efficiency in adversarial environments. Among individual mechanisms, Manager Oversight proves most effective for maintaining task success (61%) and factual consistency (67%), while Cross-Verification shows particular strength in reducing toxicity (41% remaining toxicity compared to 67% baseline). The Confidence Scoring mechanism, while computationally efficient (25% overhead), provides only moderate protection, suggesting that adversaries can learn to manipulate confidence estimates when they are not properly calibrated. The Voting Mechanism demonstrates solid performance across all metrics but introduces the highest communication overhead due to its consensus requirements. These results indicate that while no single mechanism provides complete protection, strategic combinations can significantly enhance system resilience. However, the substantial overhead of effective defenses raises important practical considerations for real-world deployments, where computational resources and latency constraints may limit the applicability of the most robust protective measures. This suggests that future research should focus on developing more efficient verification protocols and adaptive defense strategies that can balance security requirements with operational constraints.

Impact of Team Size and Composition

Table 4: Impact of team size and composition on adversarial effectiveness

Team Size	Composition	Task Success	Detection Rate	Sabotage Efficiency	Coordination Cost
3 Agents	S-R-W	0.38	0.41	0.72	1.15
4 Agents	S-R-A-W	0.29	0.31	0.82	1.00
5 Agents	S-R-A-W-E	0.23	0.26	0.87	1.32
6 Agents	S-R-A-A-W-E	0.18	0.22	0.91	1.58
Heterogeneous 4	Mixed Roles	0.25	0.35	0.78	1.24

The analysis of team size and composition in Table 4 reveals intriguing patterns in how organizational complexity affects security dynamics. As team size increases from 3 to 6 agents, adversarial effectiveness improves substantially—sabotage efficiency increases from 72% to 91% while detection rates decrease from 41% to 22%, indicating that larger, more complex systems provide better cover for malicious activities. This counterintuitive finding suggests that the coordination overhead and information diffusion in larger teams create more opportunities for adversaries to operate undetected. The heterogeneity analysis further illuminates this dynamic: teams with mixed specialized roles (Heterogeneous 4) show slightly better detection capabilities (35% vs 31%) but worse task success (25% vs 29%) compared to standardized 4-agent teams, indicating that while diversity may improve threat awareness, it also introduces coordination challenges that adversaries can exploit. The coordination cost metric shows a clear upward trend with team size, increasing from 15% overhead for 3-agent teams to 58% for 6-agent teams, creating a concerning security-efficiency trade-off where larger teams become both more vulnerable and more expensive to operate securely. These findings have profound implications for system design: rather than assuming that larger, more specialized teams inherently provide better security through diversity, architects should recognize the increased attack surface and monitoring challenges that come with organizational complexity. This suggests that optimal team design should balance specialization benefits against the security risks of complexity, potentially favoring smaller, well-integrated teams with robust verification protocols over larger, more fragmented organizational structures.

Long-term Adaptation and Evolution

Table 5: Adversarial adaptation over multiple interaction episodes

Episode	Task Success	Toxicity	Detection Rate	Adaptation Score	Strategy Complexity
1-10	0.29	0.45	0.38	0.52	0.61
11-20	0.21	0.58	0.29	0.67	0.73
21-30	0.16	0.69	0.23	0.78	0.82
31-40	0.12	0.74	0.18	0.85	0.89
41-50	0.09	0.79	0.14	0.91	0.94
With Defense	0.42	0.28	0.45	0.63	0.58

The longitudinal analysis in Table 5 demonstrates the alarming adaptive capabilities of adversarial agents over repeated interactions. Across 50 episodes, we observe a consistent improvement in adversarial effectiveness—task success drops from 29% to just 9%, while toxicity increases from 45% to 79%, and detection rates plummet from 38% to 14%. The Adaptation Score shows steady improvement from 0.52 to 0.91, indicating that adversaries successfully learn to exploit specific vulnerabilities in the team's collaborative patterns and communication protocols. Most concerning is the Strategy Complexity metric, which increases from 0.61 to 0.94, suggesting that adversaries evolve from using simple sabotage tactics to employing sophisticated, multi-stage attack strategies that are increasingly difficult to detect and mitigate. When defenses are introduced, the adversarial adaptation is somewhat constrained but still significant—the adaptation score reaches 0.63 with defenses compared to 0.91 without, indicating that while protective measures slow the learning process, they do not prevent it entirely. This evolutionary perspective reveals a critical vulnerability in current LaMAS: their static nature makes them increasingly susceptible to adversaries that can learn and adapt over time. The implication is that security measures cannot be static implementations but must themselves be adaptive, continuously evolving to counter new adversarial strategies as they emerge. This finding underscores the need for ongoing monitoring, periodic security updates, and potentially the incorporation of adversarial training directly into the multi-agent system's operational framework to maintain robustness against evolving threats.

Conclusion

This work has systematically exposed the critical security vulnerabilities inherent in LLM-based Multi-Agent Systems through the lens of adversarial infiltration. Our comprehensive framework demonstrates that optimized adversarial agents can achieve alarming success rates in sabotaging collaborative tasks while maintaining remarkable stealth, with our method achieving 82% success in misinformation attacks while evading detection in 69% of cases. The role-specific vulnerability analysis reveals counterintuitive findings, particularly that managerial positions—despite their oversight responsibilities—represent the most dangerous infiltration points when compromised.

The empirical evidence presented establishes several fundamental insights for the LaMAS research community: first, that traditional single-model red-teaming approaches are insufficient for multi-agent security assessment; second, that organizational complexity in LaMAS creates expanded attack surfaces rather than enhanced security; and third, that effective defense requires layered, adaptive mechanisms that balance protection with computational feasibility. Our evaluation of defense strategies shows that while combined approaches can restore 78% task success, they come at significant operational cost, highlighting the security-efficiency trade-off that must be addressed in real-world deployments.

References

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Chen, W.; Su, Y.; Zuo, J.; Yang, C.; Yuan, C.; Qian, C.; Chan, C.-M.; Qin, Y.; Lu, Y.; Xie, R.; et al. 2023. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. *arXiv preprint arXiv:2308.10848*.

Ganguli, D.; Lovitt, L.; Kernion, J.; Askell, A.; Bai, Y.; Kadavath, S.; Mann, B.; Perez, E.; Schiefer, N.; Ndousse, K.; et al. 2022. Red Teaming Language Models with Language Models. *arXiv preprint arXiv:2202.03286*.

Greshake, K.; Abdelnabi, S.; Mishra, S.; Endres, C.; Holz, T.; and Fritz, M. 2023. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. *arXiv preprint arXiv:2302.12173*.

Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; and Kamar, E. 2022. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3309–3326.

Hong, S.; Wu, X.; Chen, C.; Wang, W.; Zhang, J.; Huang, Z.; Li, X.; Yu, J.; Zhai, G.; Yun, N.; et al. 2023. MetaGPT: Meta Programming for Multi-Agent Collaborative Framework. *arXiv preprint arXiv:2308.00352*.

Hubinger, E.; Denison, C.; Mu, J.; Lambert, M.; Tong, M.; MacDiarmid, M.; Lanham, T.; Ziegler, D. M.; Maxwell, T.; Cheng, N.; et al. 2024. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. *arXiv preprint arXiv:2401.05566*.

Ji, J.; Qiu, T.; Chen, B.; Zhang, B.; Lou, H.; Wang, K.; Duan, Y.; He, Z.; Zhou, J.; Zhang, Z.; et al. 2023. AI Alignment: A Comprehensive Survey. *arXiv preprint arXiv:2310.19852*.

Jones, E.; Nematzadeh, A.; Grant, E.; Grefenstette, E.; and Hill, F. 2023. Automated Crossword Solving as a Natural Language Processing Benchmark. *Transactions of the Association for Computational Linguistics*, 11: 169–185.

Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Pérolat, J.; Silver, D.; and Graepel, T. 2017. A unified game-theoretic approach to multiagent reinforcement learning. *arXiv preprint arXiv:1711.00832*.

Li, G.; Hammoud, H.; Itani, H.; Khizbullin, D.; and Ghanem, B. 2023. CAMEL: Communicative Agents for" Mind" Exploration of Large Language Model Society. *arXiv preprint arXiv:2303.17760*.

Li, Z. 2025. Knowledge-Grounded Detection of Cryptocurrency Scams with Retrieval-Augmented LMs. In *Knowledgeable Foundation Models at ACL 2025*. Association for Computational Linguistics.

Li, Z.; and Ke, Z. 2025. Domain Meets Typology: Predicting Verb-Final Order from Universal Dependencies for Financial and Blockchain NLP. In *Proceedings of the 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, 156–164. Association for Computational Linguistics.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252.

Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; and Liu, Y. 2023. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. *arXiv preprint arXiv:2305.13860*.

Perez, E.; Huang, S.; Song, F.; Cai, T.; Ring, R.; Aslanides, J.; Glaese, A.; McAleese, N.; and Irving, G. 2022. Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned. *arXiv preprint arXiv:2209.07858*.

Qian, C.; Cong, W.; Liu, W.; Yang, R.; Li, Z.; He, J.; Zhang, D.; Luo, J.; Wei, Y.; Li, J.; et al. 2023. Communicative Agents for Software Development. *arXiv preprint arXiv:2307.07924*.

Ramchurn, S. D.; Huynh, D.; and Jennings, N. R. 2004. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1): 1–25.

Wallace, E.; Feng, S.; Kandpal, N.; Gardner, M.; and Singh, S. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2153–2162.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, S.; Jiang, E.; Liu, X.; Zhang, C.; Rao, M.; Hallacy, J.; et al. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. *arXiv preprint arXiv:2309.07864*.